

Concepts for Improved Visualization of Web Link Attributes

Dipl. Inform. Harald Weinreich
Distributed Systems Group
Computer Science Department
University of Hamburg, Germany
Tel.: +49 / 40/ 42883 2331
E-mail: harald@weinreichs.de

Prof. Dr. Winfried Lamersdorf
Distributed Systems Group
Computer Science Department
University of Hamburg, Germany
Tel.: +49 / 40/ 42883 2421
E-mail: lamersd@informatik.uni-hamburg.de

Keywords

hypertext, navigation, links, typed links, usability

Abstract

This paper discusses methods to generate and display automatically additional hyperlink information to the users of the World Wide Web.

Current Web browsers make it hard to predict what will happen if a link is being followed: users get different information than they expect, a new window may be opened, a download starts, or the destination object is just not available. Instead of giving an appropriate notification in advance, users have to follow a link, check whether the document contains the expected information, get back, try another link etc. However, usually it is possible to obtain additional hyperlink information from several sources like link anchor tags, the user's history and web servers. Furthermore, with little enhancements, Web servers may include even more additional information to the hyperlinks in Web documents. These can be displayed before users select a link to improve navigation and reduce the cognitive overhead.

In this paper several types of Web hyperlink information are listed, potential methods to present these facts are compared, the prototype implementation of the proposed concept – called by us *HyperScout* – is presented, and further developments are discussed.

1 Introduction

New users to the World Wide Web are usually impressed by the amount and richness of contents accessible on the Web. Soon after, they unfortunately have to discover that finding the right information is a challenging task, as search engines and hyperlinks often do not lead them to the information they expect.

The tremendous number of documents is only one of the reasons for usability problems of the Web. Nielsen discovered in a field study that users can even lose their orientation in a small hypertext system, if no orientation clues are given [22, pp. 133]. Several reasons have been identified why users often do not know where they are, what they did and what way to choose next.

Hypertext systems like the Web can be seen as a space, where readers can move directly from one place to another. The concepts of *orientation* and *navigation* refer to this travel metaphor. Unlike in the real world, getting an overview of the *hyperspace* is nearly impossible, as only one object is displayed at a time, which is an insignificantly small part of the overall space. Furthermore, hyperlinks often do not carry enough information to predict the referenced object, what makes it demanding to select the right way.

The *cognitive overhead* associated with choosing whether to follow a link or not has long been realized. Regarding hypertext navigation, already Conklin [9, pp. 40] characterized cognitive overhead as “the additional effort and concentration necessary to maintain several tasks or trails

at one time”. Different solutions were suggested to reduce this cognitive overhead. Two key aspects of these solutions are tools to allow an overview about the hyperspace and the enhancement of link capabilities.

Some hypertext systems provide additional hyperlink *overview maps* like the Guide System [35], or an extra *hierarchy browser* for structural navigation like HyperWave’s former client Harmony [10]. Unfortunately, Web clients do neither of the above. In fact the present concept of the Web does not inherently provide any kind of structural information (except of the file-system folder structure represented in the URL). Links are embedded in the HTML documents and have to be extracted first to create overview maps. While local overviews are still possible and several tools are available, comparable global navigation tools are not supported.

The second solution to reduce cognitive overhead in hypertext navigation is to improve the substance of links. Hypertext Systems like MUCH [38] and Sepia [35] enforce the author to set a link type and add link titles. This information is shown to the user before he selects a link, and he can get a more precise impression about the referenced document. In fact HTML 4.0 [27a] also supports link types and titles, but hardly any author uses them.

The main reason for this shortcoming is probably the poor support for these features by Web browsers and editors. But hypertext research did also show that setting link types is not easy. This was discovered by Wang and Rada [38] in a two-year field study with over 200 users of the MUCH hypertext system. When writing documents and setting hyperlinks, the users rarely bothered to select link types or assigned link names other than the default ones. If a link type was chosen, the choice was often inconsistent with the way other authors would categorize the link. As a result it was recommended to make specifying the link type as easy as possible.

2 Concept

Web usage studies by Catledge and Pitkow [7] or Tauscher and Greenberg [34] indicate that about half the navigation actions of users are open URL events. The GVU Survey [12] indicates that users see hyperlinks from Web pages as the most important way to find new WWW pages.

These findings show that links are the Web’s most important navigation facility. However, the user interface for hyperlinks is quite abstract: the only information a user gets about a link destination is the *link anchor text* and the *referenced URL* displayed in the status bar of browsers, like Netscape Communicator or Microsoft Internet Explorer.

Usability studies by Spool et al. [31] bore out that Web links are usually not sufficient for the user to predict the referenced document. Especially embedded hyperlinks, only one or two words long, are often too generic. Users were more certain about what link to choose when the links were displayed in a separate line and explained with an extra description. However, this is rarely available.

Missing link information has the consequence that users try to help themselves by interpreting the only additional information they have: the URL. We noticed in *thinking-aloud tests*¹ with our campus wide information system that more experienced users learn to interpret some of the information included in the URL [39]. It was apparent from the users’ comments that the URL was often an important element for the users’ orientation and navigation actions. A study on how people make link selections was conducted by Stanyer and Procter [32]. It confirmed that users are often forced to fall back on heuristics drawn from past experience and interpret the URL before selecting a link. The address was used to predict different aspects of the referenced object, like their contents, the expected download time, and the destination Web site.

¹ This method from cognitive psychology was first used for software evaluation by Lewis in 1982 [17].

However, the readability of URLs for humans is poor as they are primarily technical addresses. Particularly less experienced users are unable to understand URLs and do not know how to interpret the different parts. Consequently, several projects aim to make URLs avoidable, like RealNames² a service that maps key phrases to Web pages. Since the location of a Web object is defined by the URL, it will not become obsolete in the near future. We could still try to make it less important, also by providing the user a better link interface.

If we want to make URLs less considerable, more meaningful information has to be provided. However, hypertext research did show that authors often have problems adding useful information to hyperlinks, and actually in the Web they are rarely offered (see introduction). Therefore it might be reasonable to display that piece of information to users which is already available and might be extracted automatically.

There are several sources the browser could use to give Web links additional information. Link anchors and the referenced URLs are two of them. Like experienced users know how to interpret URLs, the browser might decode this data and present it in a more legible way. This would reduce cognitive overhead and avoid the need of the users to observe the address in the status bar when looking for information on the Web.

Another prosperous source is the user's history. By maintaining a considerably comprehensive log of the user's visited documents and offering better history tools, retrieving information could be made much easier. Moreover the user's history could be integrated more seamlessly into the navigation. Facts like the last visit to a Web page or the title of a document could be displayed to a link.

The fourth source of additional hyperlink information could be Web servers. By regularly gathering information about all local and directly referenced external documents into a database, this information could be used to add new attributes to link tags.

Since the user interface for links has hardly changed from the first versions of NCSA's Mosaic and as many users complain that they have significant problems finding things on the Web [12], there is a big potential for improvements, and we should give the users all the help we can. We try to go one step in this direction with our tool *HyperScout*.

3 Methods to Display Additional Link Information

To determine how to display additional link information most appropriately, we compared different methods introduced by earlier hypertext systems. This paragraph will give a short survey of the most promising ways to present supplementary link type data.

3.1 Overview Map

Some hypertext systems with typed links, such as SEPIA [35] and MacWeb [20], display the link information as labels within an overview diagram of the hyperspace. Labels appear next to the arrow representing the link. The advantage of this method is an improved overview of the structure and the document relations.

However, if the user wants to consider this information, he has to keep two areas in view, and the overview map needs certain additional screen space.

² The RealNames Corporation can be found at <http://www.realnames.com/>. Their "Internet Keyword Concept" has been integrated in some search engines and browsers like the Microsoft Internet Explorer 5.

3.2 Reserved Area

The most popular Web browsers display the referenced URL of a link in the browser's status bar at the bottom of the window. *Hyperties* uses the last screen line to show a short description when the cursor touches a hyperlink [19; 29].

If a system shall put more link information on view than a single line of text, an extra window, or frame can be used. For example, the hypermedia version of the papers of the August 1995 issue of "Communications of the ACM" displays additional link data in an extra frame.

Though, placing the comment so far away from its associated anchor makes it difficult to compare them. It forces the reader to make large visual saccades in order to read the comment and then return to the anchor's text.

3.3 Next to the Links

Web links sometimes provide short supplementary information next to the anchor text. Web design guidelines recommend this for links to objects with considerable properties [16; 18]. For instance, authors put the size of the referenced object in brackets after the link, if the file is reasonably big. It is also possible to use small icons to envision some kinds of link types.

However, adding such kind of data by the browser or a navigation tool automatically can cause problems with some Web documents, as even little changes can mess up the layout of the page. The Web browser tool "Traffic-Lights", which adds little colored bars before and behind every link anchor, demonstrates this [6].

3.4 Link Color

Already the first graphical Web Browser Mosaic used two different colors for the link anchor text to distinguish between links to unknown documents and links to recently visited documents. HyperWave's former client Harmony additionally utilizes different anchor text background colors to make multiple link destinations in one anchor visible.

The color of a link is an important navigation aid for Web users. Unfortunately the complexity and number of the displayed link properties is quite low. Furthermore it does only work with text link anchors, and it requires that Web designers do not change the standard colors of the documents.

3.5 Mouse Pointer

Another method to make different link attributes apparent is utilizing different mouse pointers. The pointer changes according to the link type if it is moved over a link. For example, the Guide system uses this method [23, p. 57].

This method allows to display link attributes instantly if a the pointer hovers over a link. Since it is completely independent of the screen layout, different mouse pointers may be an interesting option for Web clients, too. The only drawback is that the complexity of the displayed properties is limited, e.g. no text may be displayed.

3.6 Popup

Popups or *rollovers* are small floating windows which appear next to the mouse cursor if it is moved over an active object. Recent versions of Microsoft's Internet Explorer display the *title* attribute of anchor tags in a little popup. Jakob Nielsen strongly recommends using these titles as they "are one of the first enhancements to the Web that actually help people navigate (as opposed to simply making pages look more fancy)" [24]. Unfortunately, they can rarely be found.

Popup information is also used in most current GUI-Systems. For instance, they appear if the mouse pointer floats over Icons and control elements. That makes them quite common to users.

Nevertheless, there is a well-known disadvantage: Popups can cover some important information as they obstruct objects near to the link anchor. This problem is reduced by letting them appear only on demand when a user points the mouse over an active object for a certain time.

We chose popups as the method for our project HyperScout in order to display additional link information. They seem to be best applicable to the Web and appear most advantageous. In particular, they do not interfere with the page layout, and they even can be used together with image-maps, which hold several links in one graphic. Furthermore, they appear next to the focus of attention and can show a flexible amount of information.

4 Characteristics of Hyperlinks in the WWW

On the first view Web hyperlinks might seem simple compared to the typed links of other hypertext systems. Looking closer, there are several new features that make the use of the Web even more challenging than other hyper-documents. Additional aspects regarding Web links may be identified, like a local versus an external link. Furthermore, a lot of unexpected things can happen if a user clicks on a link: WWW clients incorporate different protocols, like news and ftp, links can control the contents of other frames or windows, or the user may be requested to write an e-mail. It also occurs that users return to pages they visited before but afterwards cannot remember their former actions.

This section describes the most important characteristics of Web hyperlinks and explains which information we think should not be hidden from users any more.

4.1 Semantics

A link between two documents sets them in a semantic relationship. Web authors try to express this relationship and give an impression of the target by selecting an appropriate link anchor text or graphic. However, in many cases this is not sufficient. Therefore semantic information like link titles and link types were introduced [4].

Thüring, Hannemann and Haake list *semantic link information* as their first principle of useful hypermedia system design [35]. Already in HTML 2.0 additional anchor attributes were defined to express the relationship between source and destination more precisely [3]. Web authors can set a link *title* and use the two attributes *rel* and *rev* to set a forward and backward relationship type. HTML 4.0 identifies several relationship types as useful like “contents”, “subsection”, or “alternate” [27b]. The approach for the prospective XLink [11] is similar, but even the HTML attributes are hardly considered in today’s browsers. Only the latest versions of the Internet Explorer display the link title in a popup window.

Until these attributes are applied by more authors, semantic information about the target document, like its title or author, is a useful source (Figure 1). This information can either be extracted from the user’s history or be transferred from the server as will be shown later.

4.2 Navigation Direction

Direction is an essential aspect of hypertext navigation. From the user’s view, it can be distinguished between forward and backward navigation. Forward navigation occurs if a user is looking for new information, whereas backward navigation occurs when he tries to regain information obtained before [35]. The most important tool for backward navigation is the back button, followed by the Bookmarks (also called Favourites or Hotlist) and the History [7; 34].

Another aspect of backward navigation is the revisit of Web pages. In a 6 weeks’ study with 23 users Tauscher and Greenberg found that about 60% of all pages an individual visits are pages he has visited before [34]. Given these statistics, we believe that Web browsers should regard a user’s history better in his active navigation. Contemporary Web browsers consider the user’s

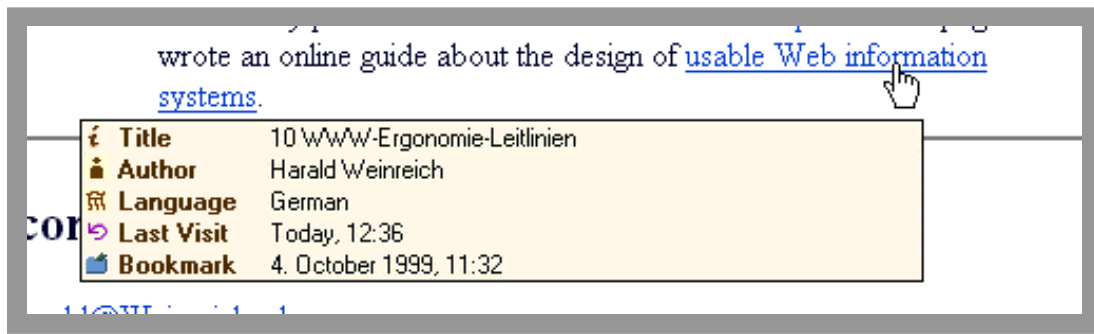


Figure 1: Link to a page the user visited before. He has also bookmarked that URL.

history only in a very unsatisfactory way: Links change their color; links to pages that have not been seen by the user are blue, links to previously seen pages are purple.

This was perhaps appropriate for the first Web browser, but it is not any more. Today many link anchors are graphics, so this information is not displayed at all. In other cases Web authors change the link colors for design purposes. Even if none of this is done, users cannot see exactly when they visited a destination object before. Furthermore the link color returns to blue after a certain time.

Bookmarks are a further aspect of backward navigation that is not considered either. A user does neither see, if he has a bookmark to a document, nor if a link leads to a page he did bookmark before. Seeing that many users have more than 100 Bookmarks [1; 12] it is unlikely that they are aware of all of them.

By displaying the information gathered from the user's history (Figure 1), we want to seamlessly integrate the user's navigation direction into active Web operation.

4.3 Access Time

System response time is a crucial factor for the usability of interactive systems [30, p. 297]. User studies with one of the first hypertext systems, the ZOG system of the Carnegie Mellon University, indicated that a maximum response time of 2s should not be exceeded [28, pp. 31].

Unfortunately a comparable response time cannot be guaranteed to the Web user. The access time to different Web objects varies quite a lot. Johnson argued that the value of an information is dependent on the access time [13], and usability test showed that a lot of users quit the transfer and return to the last object if the download takes too long [31]. Slow access time interrupts the user's task solving process, and leads to more cognitive burden and lower productivity.

The predictability of a system's response time of a link can help users to choose whether it may be worth to follow a link or not. The already mentioned Web browser tool "Traffic-Lights" uses small red, yellow and green bars next to the link anchors to give the user an idea of the anticipated transfer time. It is estimated by requesting the head of a referenced server's homepage. A user evaluation showed that this concept was beneficial for users and helped them to find suitable documents to a topic more quickly [6].

Unfortunately the access time to a Web object cannot really be predicted as there is no quality of service support in the current TCP/IP implementation. However, a big file size or a slow server response can give a good impression about the probable response speed and should be displayed to the user (Figure 2).

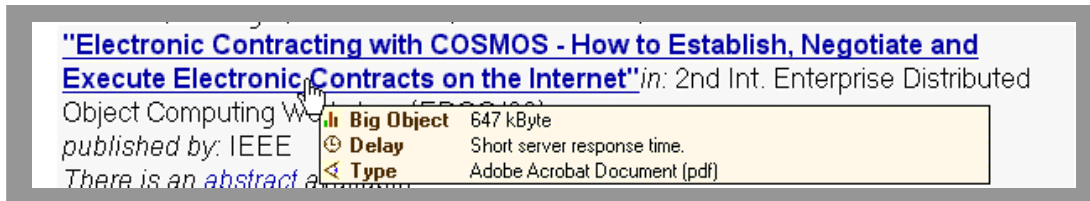


Figure 2: Link to a large pdf-file

4.4 Type of Referenced Object

Although HTML is still the primary language for documents on the Web, hyperlinks may also lead to documents in other formats. Graphics, Adobe Acrobat documents, VRML worlds and XML documents are only a selection of what can be found. Some of these types do not provide hyperlinks; others require a special plug-in or viewer. This can apparently set hurdles to the user.

Web design guidelines like [16] or [18] advise to make this somehow visible to the user. This is not always followed. Fortunately the document type can often be extracted from the extension of the referenced object (Figure 2). Nevertheless Web browsers do not make this fracture in navigation clear. The users have to look at the URL in the status bar and to learn a lot about different file types.

4.5 Link Action

Regarding the Web, hyperlinks behave in different ways. The most frequent ones are *replacement links* where the destination document replaces the source object. *Reference links* lead to another position in a Web document [22, pp. 96]. This information can still be extracted by the experienced user from the link URL, but Web links can also lead to more unpredictable actions. Some links open a new browser window, control the contents of another frame or start a new client, like an e-mail program. Some of these link actions are defined more precisely in XLink with the two new attributes *show* and *acutate*, nevertheless, the presentation of these characteristics is not defined.

Design guidelines recommend making these aspects apparent in the link anchor or next to it. Some Web authors follow these recommendations, others do not. Anyway there is no standard.

Most of this information can be extracted just by analyzing the parameters of the anchor tag, however, no common browser does this. HyperScout makes use of this information and displays it in popups (Figure 3).

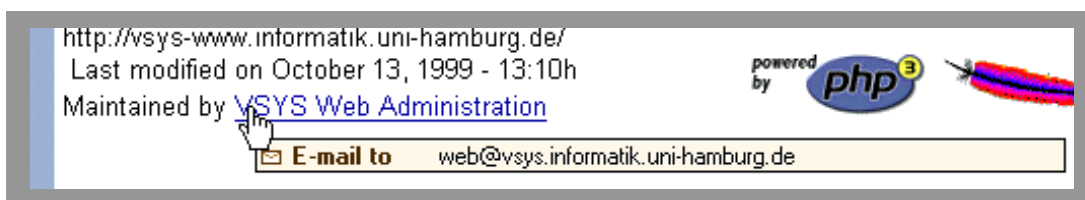


Figure 3: A "mailto:" URL

4.6 Local, Remote

Unlike a local hypertext system, hyperlinks in the Web can lead to an object from another provider than the referrer. These external links normally have another signification than internal links. Usability studies conducted by Spool et al. [31, pp. 45] pointed out that users do not expect that a link leads to a different site, if this is not illustrated. However, to the user this information is only apparent if he compares the current URL with the referenced object's address.

This cognitive overhead could easily be avoided, if Web clients would qualify external links adequately [25]. Therefore we included this attribute in our tool (Figure 4).

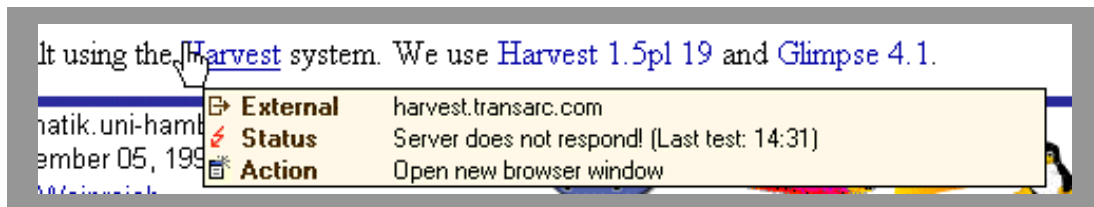


Figure 4: An external link that opens a new window. The server is down.

4.7 Link Status

Users consider dangling hyperlinks as one of the most serious usability problems of the Web [12]. A recent survey showed that nearly 6% of all Web links are broken [33]. Advanced systems like HyperWave [14] or the Atlas Link Service [26] incorporate concepts to avoid broken hyperlinks, even for a distributed hypertext, by removing them immediately when detected by the server.

However, this may lead to a new usability problem. We noticed during user tests with a prototype Web information system that users sometimes expected hyperlinks in titles of menus and in graphics that were not linked yet. Although some of those objects did not even appear like hyperlinks, the participants tried to click on them and got confused as nothing happened.

In these cases it would be more advantageous if non available links could be clearly distinguished from working links before clicking on them. If Web servers did contain a component that frequently checks all links and saves the data in a database, they could easily add this information to the transferred document's link anchor tags. Browsers could display the link status to the user and help him to find another destination until the author fixes the problem.

HyperScout has two mechanisms to check the status automatically. One works on server side and regularly checks the existence of referenced objects, the other operates along with the browser and checks servers to see if they are responding. This information is presented to the user before he is clicking on a link (Figure 4).

4.8 Further Meta Information

There are several further aspects of Web links that might be presented to users. A document's language, its popularity, or its last update time can users help to decide if to follow a link or not.

Although Web guidelines recommend making the most important and unexpected link characteristics apparent with the link anchor text, authors often do not follow this. The exact user group of a Web server can hardly be identified, therefore a Web Author can only vaguely predict what may be important clues for the user and what not.

Specifying link attributes is an additional and demanding work for hypertext authors. At least it could be reduced to some extent if Web server and browser technology would be more advanced. Especially user specific adaptable navigation tools seem promising. They have a better capability to present the user just the link information he needs. At the same time, to avoid information overload, they should only display the important data being available. This requires a configurable and situation-adaptive tool. HyperScout does not reach this objective completely yet, but it is a step in this direction.

5 Implementation

We had two options to enhance the way Web links are being displayed to the user without changing the documents on the server. Either an open-source Web browser like Mozilla³ can be

³ The Mozilla open-source Web browser project can be found at <http://www.mozilla.org/>

used and modified in the desired way, or a proxy can be used to alter the documents when they are transferred from server to client. Although the former method gives better control over the link presentation, the latter seemed more appropriate for our needs. First of all such a proxy technology can be used on client and on server side. The client proxy is used to let the browser display user specific or even group specific link information. The server side proxy is used to append additional information to the links on server side, like the status, avoiding that the client needs to pre-fetch information about the referenced documents. Furthermore the client proxy concept can be collaboratively used in a work group.

Another reason for this decision was that we primarily wanted to implement and evaluate a proof of concept prototype, and this method seemed less complicated. The capabilities of Dynamic HTML are sufficient for our concept as will be described later.

For the implementation we used an expandable proxy framework by IBM alphaworks called WBI (Web Browser Intelligence). Barrett and Maglio describe this technology as a new approach to programming Web applications [2]. They call the concept *intermediary*, an new place for producing and manipulation Web data in several ways. WBI is freely available in Java for researchers⁴.

To parse, analyze and modify the documents in the intermediary we used Arthur Do's HTMLStreamTokenizer⁵. For persistent data storage the relational database system MySQL was employed⁶.

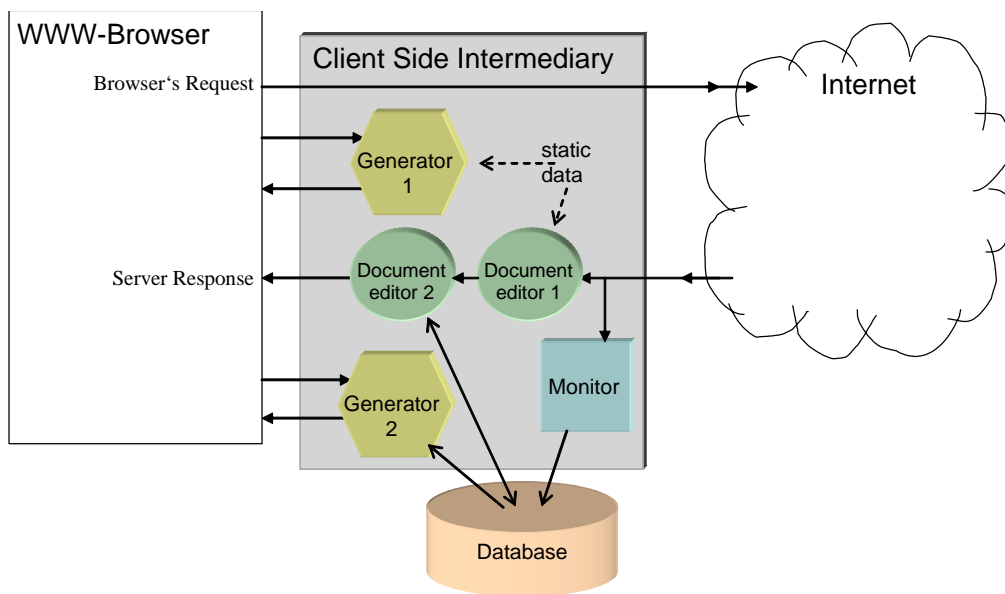


Figure 5: The architecture of the HyperScout plug-in

5.1 Client Side Implementation

The client side proxy has three tasks: to record the browsing history of the user, generate additional link information, and alter the transferred data between server and client. To accom-

⁴ The WBI Development Kit for Java is available from: <http://www.alphaworks.ibm.com/tech/wbidk>

⁵ The HtmlStreamTokenizer can be downloaded as Java source code from Arthur Do's site. <http://www.do.org/products/parser/>

⁶ The MySQL Project Homepage can be found at: <http://www.mysql.org/>

plish this, we wrote a WBI *plug-in* consisting of several components called *MEGs*⁷. They have the following structure and tasks:

Every server response is recorded by the Monitor (Figure 5). It gathers status, attributes and links of every requested document. This data is stored into the database to track the user's navigation.

Meanwhile document editor 1 adds commands to the head of the document, which tell the browser to request⁸ a short CSS definition and some JavaScript functions from the intermediary's Generator 1 (Figure 5). This code is needed to display the popups. This Generator just transfers files from the hard disk of the intermediary.

Subsequently document editor 2 parses the body of the document and adds JavaScript event commands to all anchor and image map tags (<A>- and <AREA>-tags). These events call the JavaScript functions of the header when the mouse pointer floats over a link. At the end of the document another line is inserted by document editor 2 which requests data from the intermediary's second generator. Generator 2 produces Dynamic HTML code on the fly. It consists of as many popup elements as distinct links are in the document. These elements contain the additional link information (see Figures 5 and 6).

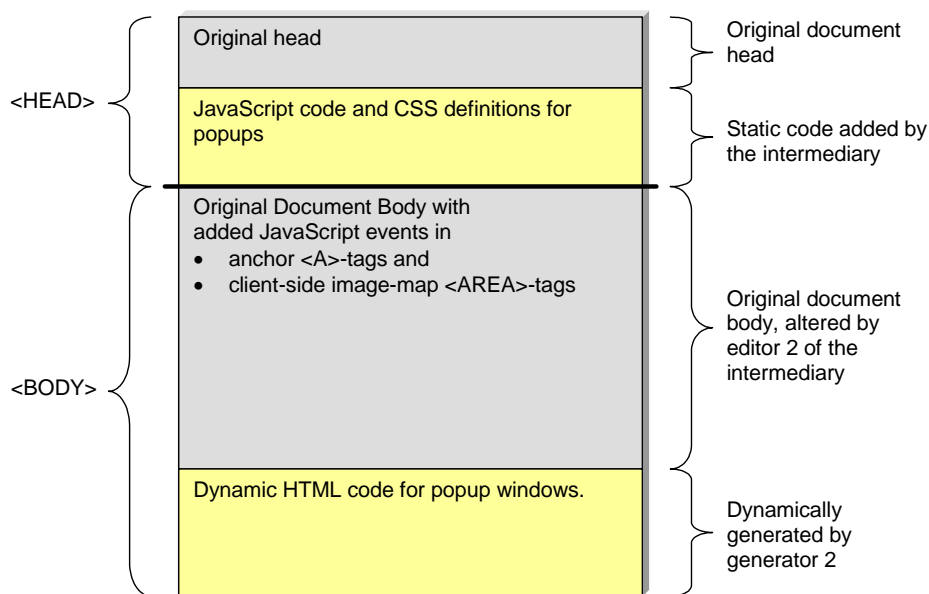


Figure 6: Document after the modification by HyperScout

Appending the code for the popups at the end of the document has the advantage that the page can be displayed by the browser before the popups are generated. This decelerates browsing the Web as little as possible.

The Dynamic HTML code concept consists of the following components: The Document Object Model (DOM) is used to address the popups, which consist of a HTML division (<div>-Tag) including a table. Cascading Style Sheets control the appearance, visibility and position of these

⁷ MEG stands for Monitor, Editor, Generator. These are the main concepts for WBI plug-in components [2].

⁸ Current Web browsers recognize commands to insert other objects into a document. This concept is related to Ted Nelson's transclusions [21]. For a CSS-definition the <LINK rel="stylesheet" type="text/css" href="popup.css"> tag is used and for JavaScript <SCRIPT Language="JavaScript1.1" SRC="popup.js">.

divisions [5; 36]. Via JavaScript the browser reacts on the user's mouse movements⁹. If the mouse pointer hovers over a link for more than 1.2 seconds the additional link information is displayed next to it. After leaving the link it disappears instantly. Moving the pointer over another link within 0.4 seconds, a new label appears instantly. Like this popups appear only "on demand".

The monitor of the HyperScout plugin has the further function, to the contact the external servers that are referred within the actual page, to estimate their response time. Another tool parses the bookmark file and adds the URLs to the database (not depicted).

5.2 Server Side Implementation

If the browser shall display additional link information about referenced objects, that data can be taken from the user's history if he visited the documents before. Otherwise it would normally be necessary, to pre-fetch all referenced objects (at least their headers) to get data on the link status, or the document size and title. However, the present situation of the Internet does not make this seem advisable as pre-loading increases the amount of transferred data significantly and therefore slows down Web usage.

We developed a solution that avoids this problem. Our concept proposes a second intermediary at the Web server. This intermediary has a database with meta information on all the documents offered by the Web server. It contains also data on all directly referenced external objects.

When a document is requested, the intermediary adds new attributes to all anchor-tags, like the title of the referenced document, its size, language, and the links status (Figure 7). For this purpose we used XHTML 1.0 to define additional tags to the anchor tag [37]. We called these attributes *hreftitle*, *hrefsize*, *hreflang*¹⁰ etc.

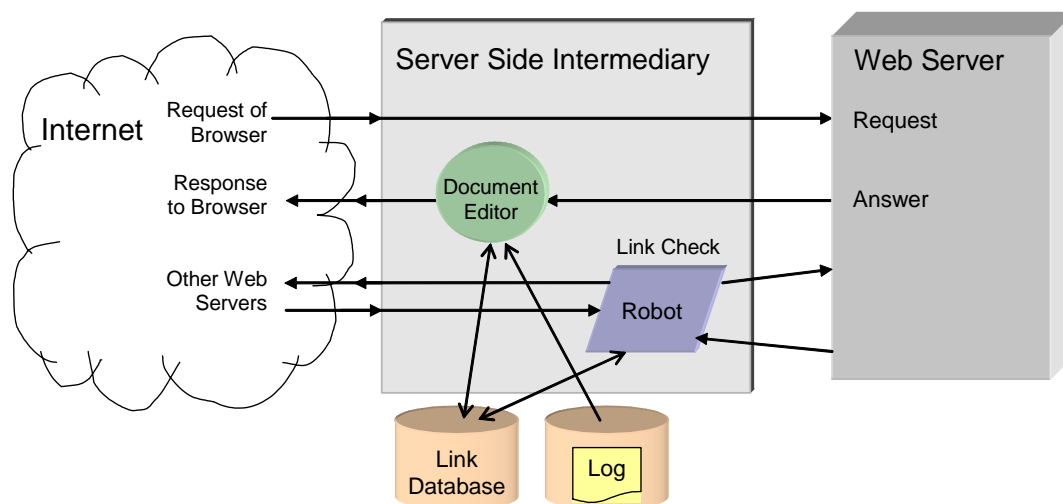


Figure 7: The structure of the server side intermediary

To check all documents and their internal and external links, a robot of the intermediary is started regularly, for instance once a week. By this means, the intermediary does not always

⁹ Although the models used by Netscape's Communicator and Microsoft's Internet Explorer to address and control HTML objects are dissimilar, it is possible to make JavaScript functions work with both browsers as of version 4.

¹⁰ In fact the attribute *hreflang* of the anchor tag was defined in HTML 4.0 [27c].

have consistent data, but most of the time; much more steadily than usual for the Web. The information is in a state of *weak consistency*.

Doubtlessly, systems which keep all documents and links in a database (e.g. HyperWave) are more advanced than ordinary Web servers. They can guarantee consistent meta and link data and could add it directly to the link tags. However, the concept we propose can also be used with any current Web Server without the need of major modification to its documents or the programmed dynamic pages (like servlets). Therefore we think it is more likely that it might be used with current Web services.

5.3 Limitations

It is a disadvantage of this implementation that the inserted Dynamic HTML code occasionally interferes with the code of the original document. However, this could be avoided with a more sophisticated implementation. More often we had problems with HTML frames that left only insufficient space for the popups.

Another weakness of the prototype is the deceleration of the page transfer. Java seems to be still too slow for this kind of application. But we can still recommend WBI for prototyping.

After all, this implementation of HyperScout is primarily an example. The listed problems speak for a direct implementation of these concepts into browser and server.

6 Future Work

The next step we will accomplish is a usability evaluation of our concept. First of all we need to know if these enhancements really help to find information faster and easier. A result could be that users follow wrong links less often and therefore need to *undo* their operation by pressing the back-button more rarely.

Several aspects of this link interface concept are new and need further research. In particular we are afraid that the dynamic character of the popups might be confusing. It is possible that beginners would prefer popups displaying information on the same attributes for every link.

We are thinking of further enhancements to HyperScout. They will provide information on a user's own navigation habits and help him to find and regain information easier. Moreover, we want to offer strategic information on the actions of other users of a Web service, which can for instance be extracted from the server log (Figure 7). Traces left by other users can help current users to find and understand information easier [40].

Furthermore, additional hyperspace information, like link traversal statistics and incoming links to a document¹¹, could be added to the documents by the server's intermediary and displayed on demand. Readers can find out how other users came to a page and discover along these lines newer documents about a subject, that link to the actual page [8].

7 Related work

This paper did already refer to several preceding projects. This section lists three recent projects with related activities.

A novel technique to display link information was presented by Zellweger, Chang and Mackinlay [41]. The concept is called *fluid links*, as the document adjusts dynamically and makes space for a "gloss", i.e., additional information to a link. The advantage of this technique is that the original document is not covered by a popup window. However, applying this

¹¹ Most Web clients transfer the URL of the preceding document (referrer) of a requested object to the Web server. This information can be stored and obtained later from the server log.

technique to today's Web pages will be quite unfeasible as little changes to HTML-documents can already mess up the entire page layout.

A prototype system presented by Kopetzky and Mühlhäuser uses a similar proxy technology to display *thumbnail* previews of the referenced documents [15]. This concept probably helps users to remember navigation paths easier as they might be able to recognize a page visited earlier. Nevertheless, this vision has disadvantages: Reduced previews of a page usually do not allow to read any of its information. Besides most sites nowadays use the corporate identity of an organization what makes the thumbnails of all pages look alike. Moreover the data transfer necessary to generate the previews would slow down the use of the World Wide Web significantly as all referenced pages including graphics would have to be pre-loaded with a document.

The project "*Link Lens*" deals with link information previews as well. Stanyer and Procter propose a tool that displays additional link information if the tool is moved over a Web link. The Link Lens pre-fetches additional data about the referenced document on demand and displays it in a kind of magnifying glass [32].

8 Conclusion

Previous hypertext research did show that additional link information helps users to get a better idea of the target of a link. Like opening every door and peering inside before entering, this can help to avoid unnecessary navigational operations and therefore reduce the cognitive overhead.

HyperScout's automatically generated additional link information can adaptively point out the most important available facts, and popups display them on demand before a user follows a link. If the frequency may be reduced that Web users select a wrong link, they could save a lot of time and find required documents earlier.

The proposed methods are also an approach to make URLs for Web navigation less important. Furthermore, the history of the user is more seamlessly integrated into his navigation, and the fight against broken links has come one step ahead.

The HyperScout concept combines novel ideas with desired features of previous approaches: additional link information is generated automatically, intermediaries monitor and enrich the transferred information, and popups are used to display this information to the user.

Our prototype implementation proves that the ideas work. Several advantages can be identified: the implementation is lightweight and almost platform independent. It works with the most popular Web clients and on any platform that supports Java. It can also be used with the majority of Web servers without the effort of Web authors to change current or future documents.

Still a lot of work has to be accomplished to improve the usability and the abilities of HyperScout.

Acknowledgements

Special thanks go to Volkert Jürgens and Matthias Mayer for their support in developing and implementing these concepts.

More information on the project and a prototype is available at:
<http://vsys-www.informatik.uni-hamburg.de/projects/hyperscout/>

References

- [1] David Abrams, Ron Baecker, Mark Chignell. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. In: Proceedings of CHI'98 Human Factors in Computer Systems, Los Angeles. ACM Press, pp. 41-48, 1998
- [2] Rob Barrett, Paul Maglio. Intermediaries: New Places for Producing and Manipulating Web Content. In: Computer Networks and ISDN Systems (Proceedings of the 7th WWW Conference), Vol. 30, pp. 509-518, 1998
- [3] Tim Berners-Lee, D. Connolly (Eds.). HTML 2.0 specification: The Achor-Element. World Wide Web Consortium, 1995
Available at: http://www.w3.org/MarkUp/html-spec/html-spec_5.html#SEC5.7.3
- [4] Michael Biber, Fabio Vitali, Helen Ashman, V. Balasubramanian, Harri Oinas-Kukkonen. Forth Generation Hypermedia: Some Missing Links for the World Wide Web. In: International Journal of Human Computer studies (Special issue on World Wide Web Usability) Vol. 47. No. 1, Academic Press, 1997, pp. 31-65
- [5] Bert Bos (Ed.): Cascading Style Sheets. World Wide Web Consortium, 1999
Available at: <http://www.w3.org/Style/css/>
- [6] Christopher Campbell, Paul Maglio. Facilitating Navigation in Information Spaces: Road-Signs on the World Wide Web, International Journal of Human-Computer Studies, 50, Academic Press, 1999, pp. 309-327
- [7] Lara Catledge, James Pitkow. Characterizing browsing strategies in the World Wide Web. In: Computer Systems and ISDN Systems (Proceedings of the 3rd International World Wide Web Conference, Darmstadt, Germany), Vol. 27, pp. 1065-1073, 1995
- [8] Soumen Chakrabarti, David Gibson, Kevin McCurley. Surfing the Web Backwards. In: A. Mendelzon et al. (Eds.). Journal of Computer Networks and ISDN Systems (Proceedings of the 8th International WWW Conference, Toronto), Vol. 31, 1999
- [9] Jeff Conklin. Hypertext: An Introduction and Survey. IEEE Computer 20 (9), pp. 17-40, September 1987
- [10] Wolfgang Dalitz, Gernot Heyer. Hyper-G – Das Internet-Informationssystem der 2. Generation. dpunkt-Verlag, Heidelberg, 1995
- [11] Steve DeRose, Eve Maler, David Orchard, Ben Trafford (Eds.). XML Linking Language (XLink). World Wide Web Consortium XLink Working Group, 2000
Available at: <http://www.w3.org/TR/xlink>
- [12] Graphic, Visualization, and Usability Center's (GVU) 10th WWW User Survey. Available at: http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/
- [13] Chris Johnson. What's the Web Worth? The Impact of Retrieval Delays on the Value of Distributed Information. Workshop on Time and the Web, Staffordshire University, 1997
- [14] F. Kappe. A Scalable Architecture for Maintaining Referential Integrity in Distributed Information Systems. IICM Graz University of Technology, Austria, 1994
- [15] Theodorich Kopetzky, Max Mühlhäuser. Visual Preview for Link Traversal on the WWW. In: A. Mendelzon et al. (Eds.). Journal of Computer Networks and ISDN Systems (Proceedings of the 8th International WWW Conference, Toronto), Vol. 31, pp. 1525-1532, 1999
- [16] Rick Levine. Guide to Web Style. Sun Microsystems, 1996
Available at: <http://www.sun.com/styleguide/>

- [17] Clayton Lewis. Using the "Thinking-Aloud" Method in Cognitive Interface Design. IBM Research Report RC 9265, IBM Watson Research Center (Yorktown Heights, New York) 1982
- [18] Patrick Lynch, Sarah Horton. Web Style Guide : Basic Design Principles for Creating Web Sites. Yale University Press, 1999
Available at: <http://info.med.yale.edu/caim/manual/>
- [19] Gary Marchionini, Ben Shneiderman. Finding Facts vs. Browsing knowledge in hypertext systems. In: IEEE Computer, Vol.21, No. 1, pp. 70-80, 1988
- [20] Jocelyne Nanard, Marc Nanard. Hypertext Design Environments and the Hypertext Design Process. Communications of the ACM, Vol. 38, No. 8, pp. 49-56, 1995
- [21] Theodor Nelson: Literary Machines. Sausalito, CA: Mindful Press, 1987
- [22] Jakob Nielsen. Hypertext and Hypermedia. Academic Press Professional, 1993
- [23] Jakob Nielsen. Multimedia and Hypertext: The Internet and Beyond. Academic Press Professional, 1995
- [24] Jakob Nielsen. Using Link Titles to Help Users Predict Where They are Going. Jakob Nielsen's Alertbox, 11. January 1998
Available at: <http://www.useit.com/alertbox/980111.html>
- [25] Jakob Nielsen. Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis, 2000
- [26] James Pitkow, R. Kipp Jones. Supporting the Web: A Distributed Hyperlink Database System. In: Computer Networks and ISDN Systems (Proceedings of 5th International World Wide Web Conference, Paris, France). Vol. 28, pp. 981, 1996
- [27] Dave Raggett, Arnaud Le Hors, Ian Jacobs (Eds.). HTML 4.0 specification. World Wide Web Consortium, 1998
a) Available at: <http://www.w3.org/TR/REC-html40>
b) HTML Link Types: <http://www.w3.org/TR/REC-html40/types.html#type-links>
c) The A Element: <http://www.w3.org/TR/REC-html40/struct/links.html#edef-A>
- [28] G. Robertson, D. McCracken, A. Newell. The ZOG Approach to Man-Machine Communication. Department of Computer Science, Carnegie Mellon University (Pittsburgh) 1979
- [29] Ben Shneiderman, G. Kearsley. Hypertext Hands-On: An Introduction to a New Way of Organizing and Accessing Information. Addison Wesley, 1989
- [30] Ben Shneiderman: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Second edition. Addison Wesley, 1992
- [31] Jared Spool, Taran Scanlon, Will Schroeder, Carolyn Snyder, Terri DeAngelo. Web Site Usability: A Designer's Guide. Morgan Kaufmann Publishers, 1999
- [32] Dominic Stanyer, Rob Procter. Improving Web Usability with the Link Lens. In: A. Mendelzon et al. (Eds.). Journal of Computer Networks and ISDN Systems (Proceedings of the 8th International WWW Conference, Toronto), Vol. 31, pp. 455-466, 1999
- [33] Terry Sullivan. "All Things Web", May 1999 Available at:
<http://www.pantos.org/atw/35654.html>
- [34] L. Tauscher, S. Greenberg. How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. In: International Journal of Human Computer Studies (Special issue on World Wide Web Usability), Vol. 47, No. 1, pp. 97-138, 1997

- [35] Manfred Thüring, Jörg Hannemann, Jörg Haake. Hypermedia and Cognition: Designing for Comprehension. Communications of the ACM, Vol. 38, No. 8, pp. 57-66, 1995
- [36] W3C DOM Working Group: Document Object Model Level 1, 1999
Available at: <http://www.w3.org/DOM/>
- [37] W3C HTML Working Group. XHTML 1.0: The Extensible HyperText Markup Language, 1999
Available at: <http://www.w3.org/TR/xhtml1/>
- [38] Weigang Wang, Roy Rada. Experiences with Semantic Net Based Hypermedia. In: International Journal of Human Computer Studies, Vol. 43, pp. 419-439, 1995
- [39] Harald Weinreich. Ergonomie von Hypertext-Systemen und das World Wide Web: Evaluation und Überarbeitung des WWW-Informationssystems des Fachbereiches Informatik. Diploma Thesis at the Computer Science Department of the University of Hamburg, 1997
- [40] Alan Wexelblat. History-Based Tools for Navigation. In: Proceedings of the Hawai'i International Conference on System Sciences, IEEE Press, 1999
- [41] Polle Zellweger, Bay-Wie Chang, Jock Mackinlay. Fluid Links for Informed and Incremental Link Transitions. In: Proceedings of ACM Hypertext'98 (Pittsburg), pp. 50-57, 1998

Vitae



Harald W. R. Weinreich studied computer science at the University of Hamburg (Germany) and the University of Limerick (Ireland). After obtaining his diploma degree in 1998, he became a research assistant and Ph.D. student at the Research Group on Distributed Systems (VSYS) of the Computer Science Department of the University of Hamburg. His research interests are distributed information systems, hypertext systems, Web usability, E-Commerce and usability engineering.



Winfried Lamersdorf holds a full professorship in the Department of Computer Science of the University of Hamburg and leads the Research Group on Distributed Systems. Areas of his current research include application-layer communication, system support for open distributed systems, open distributed software architectures and applications, distributed co-ordination and co-operation, and specific distributed application areas, such as Electronic Libraries and Electronic Commerce. Since 1986, he has been involved actively in various national and international standards bodies in the area of Open Systems and Applications. He organized and led various national and international research projects.